Model-Driven Design (aka Model-Driven Systems Design, Model-Driven Architecture)

Initially assembled by David Bernholdt (ORNL) based on email exchanges primarily with Holly Takamura (IBM), Kevin Brown (IBM), and Andy Loebl (ORNL). Corrections and additions welcome from all.

Specifications

- SysML
- UML

Books, Tutorials, etc.

- book: A Practical Guide to SysML: The Systems Modeling Language
- tutorial: OMG System Modeling Language (OMG SysML) Tutorial
- tutorial: Simulation-Based Design Using SysML?Part 1: A Parametrics Primer

IBM Links

- MDSD Redbook
    - Abstract: This IBMA(R) RedbooksA(R) publication describes the basic principles of the RationalA(R) Unified ProcessA(R) for Systems Engineering, which is IBM Rationala**s instantiation of model-driven systems development (MDSD). MDSD consists of a set of transformations that progressively refine knowledge, requirements, and design of complex systems. MDSD begins with activities and artifacts meant to promote an understanding of the system's context. Requirements problems often arise from a lack of understanding of context, which, in MDSD, means understanding the interaction of the system with entities external to it (actors), understanding the services required of the system, and understanding what gets exchanged between the system and its actors. Managing context explicitly means being aware of the shifts in context as you go from one model or decomposition level to the next. MDSD suggests that a breadth-first collaboration based approach across multiple viewpoints is more effective than a traditional depth-first functional decomposition in creating an architecture that will not only meet requirements, but will prove to be more resilient in the face of inevitable change. MDSD also seeks to provide an effective distribution of responsibilities across resources. Joint realization and abstractions such as localities provide an effective and elegant way of accomplishing this. Finally, the ability to attach attributes and values to modeling entities and the parametric capabilities of SysML provide a basis for doing simulations or other models to meet cost, risk, and other concerns.
- IBM DeveloperWorks: An overview of the Systems Modeling Language for product and systems development
    - Part 1: Requirements, use-case, and test-case modeling
    - Part 2: Structuring the Rain Sensing Wiper system
    - Part 3: Modeling system behavior
- Link to IBM Rational partner EmbeddedPlus SysML simulation toolkit 'Simulation Toolkit for the IBM Rational Software Development Platform'
    - Excerpts:

- The simulation capability allows the dynamic behavior of the system to be modeled and observed during execution, including the automatic generation of sequence diagrams capturing the interactions between system entities.
- The simulation frameworka**s timing service allows for large scale simulations to be run irrespective of actual time, allowing for analysis of large scale systems in acceptable periods of time.

Andy Loebl Links

- Telelogic (now part of IBM Rational) webinars are often useful and interesting. A different set of tools than the main Rational product line, but the same concepts.
- presentation: System(s) Architecture Practices

Tools

- IBM Rational Software Architect (SysML, UML)
- Papyrus plugin for Eclipse (SysML, UML)